# LEGIBILITY NOTICE

A major purpose of the Technical Information Center is to provide the broadest dissemination possible of information contained in DOE's Research and Development Reports to business, industry, the academic community, and federal, state and local governments.

Although a small portion of this report is not reproducible, it is being made available to expedite the availability of information on the research discussed herein.

LA-UR 89-15

LA-UR--89-15

DE89 006552

# STABLE MATRIX-MULTIPLICATION ALGORITHMS FOR LOW TEMPERATURE NUMERICAL SIMULATIONS OF FERMIONS


AUTHOR(S): F. Y. Loh, Jr.
T. E. Gubernatis
* R. T. Scalettar
** R. L. Sugar
** S. R. White


SUBMITTED TO: Plenum Press, New York


*Department of Physics
University of Illinois
Urbana, IL 61801

**Department of Physics
University of California
Santa Barbara, CA 93106





Los Alamos National Laboratory
Los Alamos, New Mexico 87545

# STABLE MATRIX MULTIPLICATION ALGORITHMS FOR LOW TEMPERATURE NUMERICAL SIMULATIONS OF FERMIONS

E. Y. Loh, Jr. and J. E. Gubernatis

Theoretical Division
Los Alamos National Laboratory
Los Alamos, NM 87545

R. T. Scalettar

Department of Physics
University of Illinois
Urbana, IL 61801

R. L. Sugar and S. R. White

Department of Physics
University of California
Santa Barbara, CA 93106

## INTRODUCTION

In this note, we discuss the use of matrix factorizations to stabilize the numerical matrix multiplications and inversions needed to simulate systems of interacting fermions at low temperatures. While the essence of a stable numerical algorithm is presented, we mainly emphasize the concepts of stabilization. A detailed description of a ground state and finite temperature algorithm is given elsewhere [1]

In order to perform a numerical simulation of fermion systems in more than one dimension, it is necessary to integrate out the fermion degrees of freedom. This ordinarily requires the use of a Hubbard Stratonovich transformation, which can be either discrete[2] or continuous. This transformation reduces a system of self interacting fermions to one in which the fermions interact only with a time varying external field determined by the Hubbard Stratonovich (HS) fields. Since the problem now has no self interactions, the fermion degrees of freedom are integrated away analytically[3], leaving one with a sum or integral over the HS field's which can be performed by Monte Carlo[3], hybrid molecular dynamics[4] or Langevin[5] methods.

At finite temperatures all these methods for performing fermionic simulations require the matrix elements of the equal time Green's function

$$\mathbf{G}(\tau,\tau) = [\mathbf{I} + \mathbf{B}_\tau \mathbf{B}_{\tau-1} \cdots \mathbf{B}_2 \mathbf{B}_1 \mathbf{B}_{N_\tau} \cdots \mathbf{B}_{\tau+1}]^{-1} \qquad (1)$$

where each $N \times N$ matrix $\mathbf{B}_\tau$ is a single particle propagator on a lattice of $N$ sites from imaginary time $\tau = N_\tau \Delta\tau$ to $\tau$ and the imaginary time path integral has been

discretized in units of the "Trotter" parameter $\Delta\tau$. Given $\mathbf{G}(\tau,\tau)$, Blankenbecler *et al.*[3] give simple rules for updating the HS fields, for correcting $\mathbf{G}$ when the HS fields are changed, for quickly calculating $\mathbf{G}(\tau+\Delta\tau,\tau+\Delta\tau)$, and for measuring physical observables at imaginary time $\tau$. Unfortunately, as $\beta$ becomes large, the matrices needed to produce $\mathbf{G}(\tau,\tau)$ become ill conditioned so the Green's function cannot be computed reliably. The reason is that as many $\mathbf{B}$ matrices are multiplied together, small-scale, high energy states get buried by the large scale, low energy states. The high energy states are not lost, rather, one formally calculates Slater determinants, taking differences of large scale matrix elements to pick up the small scale, high energy features of the band. This is a hopelessly noisy procedure that is intolerable for fermionic systems since for a typical HS configuration, information about all energy scales is needed to accurately compute the fermion matrices. Conventional simulations ultimately fail simply because these the important small scale features cannot be pulled out of $\mathbf{B}_\tau \cdots \mathbf{B}_{\Delta\tau}$ using computers with finite precision. Recents efforts[6,7] at low temperature stabilization use matrices of increased dimension, at a substantial cost of computer time and memory.

## STABILIZATION METHODS

To protect important, small scale features present in the product of the $\mathbf{B}$ matrices, we take our clues from basic texts on numerical analysis.[8] The *condition number* of a matrix is roughly the ratio of the largest singular value of the matrix to the smallest one and represents an upper bound to the amplification of errors in matrix multiplications. The realized amplification may, in fact, be much less. Multiplication of two diagonal matrices, for example, is very stable no matter how ill conditioned the factors may be. Meanwhile, orthogonal matrices are also stable multipliers since their condition number is one: there is absolutely no variation in scales. Thus, orthogonal and diagonal matrices figure prominently in matrix factorization.

Our aim then is to decompose ill conditioned matrices by representing them in the form $\mathbf{UDV}$, where the diagonal matrix $\mathbf{D}$ contains the diverging singular values and has the large condition number but where $\mathbf{U}$ and $\mathbf{V}$ are "sufficiently well conditioned," a property which will be made more precise later. If we choose both $\mathbf{U}$ and $\mathbf{V}$ to be orthogonal, the resulting decomposition is the singular value decomposition (SVD), which is known to be very stable. Unfortunately, SVD decompositions are relatively slow to perform on a computer, so in practice we use the modified Gram Schmidt (MGS) factorization $\mathbf{UDV}$, where $\mathbf{U}$ is orthogonal and $\mathbf{D}$ is diagonal but $\mathbf{V}$ is unit triangular. We found the time form a MGS decomposition is in some cases up to 20 times less then that for SVD. Fortunately, we also found that the unit triangular matrices are "sufficiently well conditioned" so we can multiply large numbers of them in our simulations without destroying stability.

To stably compute the product of many matrices, we decouple the various scales present throughout the calculation. To illustrate what we mean, we first imagine we have decomposed a partial product of the $\mathbf{B}$ matrices into the form $\mathbf{UDV}$. To multiply several more $\mathbf{B}$ matrices on the left, we then write

$$\mathbf{B}\cdots\mathbf{BUDV} = (\mathbf{B}\cdots\mathbf{BUD})\mathbf{V} = (\mathbf{U}'\mathbf{D}'\mathbf{V}')\mathbf{V} = \mathbf{U}'\mathbf{D}'(\mathbf{V}'\mathbf{V}) \tag{2}$$

giving the decomposition of the next partial product. Again, the $\mathbf{V}$ matrices must be sufficiently well conditioned that we can multiply many of them together stably. We can do this for both orthogonal (for SVD) and unit triangular (for MGS) $\mathbf{V}$ matrices.[9] The other assumption in (2) is that the decomposition of $\mathbf{B}\cdots\mathbf{BUD}$ into $\mathbf{U}'\mathbf{D}'\mathbf{V}'$ can be performed stably. This is in fact possible since for a limited number of $\mathbf{B}$'s, the diverse scales in $\mathbf{B}\cdots\mathbf{BUD}$ are well separated in different columns.

The most obvious way of stabilizing a fermionic simulation is simply to replace the straightforward computation of the Green's function by a computation which handles decomposed matrices. One may build up the decomposition for

$$\mathbf{B}_r \cdots \mathbf{B}_{r-\Delta r} = \mathbf{U}\mathbf{D}\mathbf{V}$$

by generating successive partial products in the manner described above. Then, to stablize the matrix inversion, we follow the steps

$$\begin{aligned}\mathbf{G} &= (\mathbf{1} + \mathbf{B}_r \cdots \mathbf{B}_{r-\Delta r} \cdots)^{-1} = (\mathbf{1} + \mathbf{U}\mathbf{D}\mathbf{V})^{-1} = \mathbf{V}^{-1}\mathbf{U}^{-1}(\mathbf{V}^{-1} + \mathbf{D}\mathbf{U}^{-1})^{-1} \\ &\quad \mathbf{V}^{-1}(\mathbf{U}'\mathbf{D}'\mathbf{V}')^{-1}\mathbf{U}^{-1} = \mathbf{V}^{-1}\mathbf{V}'^{-1}\mathbf{D}'^{-1}\mathbf{U}'^{-1}\mathbf{U}^{-1}\end{aligned} \tag{3}$$

These steps isolate as long as possible the matrices which contain the bulk of the scale variations, and the ill conditioned sum $(\mathbf{U}^{-1}\mathbf{V}^{-1} + \mathbf{D})$ is inverted by first decomposing and separating scales. Thus, we only ever need to invert orthogonal, diagonal, and triangular matrices, all of which are easy to invert. The factors in the final expression in (3) can be multiplied in any order.

It is useful to remark that although the effect of the HS transformation was to remove interactions among the fermions, they still interact with a time dependent external field and that the time dependence of this field reduces computational options. If the field were time independent, we could rewrite (3) as

$$\mathbf{G} = (\mathbf{1} + \exp(-\beta\mathbf{H}))^{-1} = \mathbf{S}(\mathbf{1} + \mathbf{D})\mathbf{S}^{-1} \tag{4}$$

where $\mathbf{S}$ is the similarity transformation that diagonalizes the Hamiltonian $\mathbf{H}$. In some sense, we are "diagonalizing" the problem, but we are not using a similarity transformation. In (4), we also remark that the elements of $(\mathbf{1} + \mathbf{D})^{-1}$ are $1/(1 + \exp(-\beta(E_k - \mu))$ in which terms, 1 and $\exp(-\beta(E_k - \mu)$, of very different scales at low temperatures are combined, but the effects of the small eigenvalues of $\exp(-\beta\mathbf{H})$, the $\exp(-\beta(E_k - \mu)$, are naturally cut off by their addition to 1 so they are unable to corrupt the inversion. Similarly, in (3), where we have prolonged the combination of scales as much as possible until the last step, $\mathbf{U}\mathbf{V}^{-1}$ cuts off the scales in $\mathbf{D}$.

## OTHER APPLICATIONS

An alternative to the great deal of recomputation involved in re evaluating (3) for many different $r$ is to use computer memory to store partial products. Now, we imagine we have the decompositions of *two* partial products:

$$\mathbf{B}_r \cdots \mathbf{B}_{\Lambda r} = \mathbf{U}_R\mathbf{D}_R\mathbf{V}_R$$

and

$$\mathbf{B}_l \cdots \mathbf{B}_{r+\Delta r} = \mathbf{V}_l\mathbf{D}_l\mathbf{U}_l$$

Then we can write

$$\begin{aligned}\mathbf{G} &= (\mathbf{1} + \mathbf{U}_R\mathbf{D}_R\mathbf{V}_R\mathbf{V}_l\mathbf{D}_l\mathbf{U}_l)^{-1} \\ &\quad \mathbf{U}_l^{-1}(\mathbf{U}_R^{-1}\mathbf{U}_l^{-1} + \mathbf{D}_R\mathbf{V}_R\mathbf{V}_l\mathbf{D}_l)^{-1}\mathbf{U}_R^{-1}\end{aligned} \tag{5}$$

Again, the inverse of the ill conditioned sum can be stabilized by decomposing the sum and then inverting its individual pieces. Because we kept the diagonal matrices on the outsides of the terms, elements of different scales are added together only to "cut scales off", as before. To implement (5), one needs to compute the decompositions $\mathbf{U}_R\mathbf{D}_R\mathbf{V}_R$ and $\mathbf{V}_l\mathbf{D}_l\mathbf{U}_l$ of the partial products $\mathbf{B}_r \cdots \mathbf{B}_{\Lambda r}$ and $\mathbf{B}_l \cdots \mathbf{B}_{r+\Delta r}$ from scratch only once, at the beginning of the program. Thereafter,

of a few additional factors of $\mathbf{B}$ to one partial product using (2) and the deletion of these factors from the other partial product by recalling a previously stored partial product. While the memory needs of this algorithm are considerable, they are affordable.

For calculating zero-temperature, equal-time correlation functions, Sorella, *et al* [3] have shown many benefits of projecting the ground state out of a trial wavefunction $|\psi_0\rangle$ using $\exp(-\beta\mathbf{H})$,

$$\langle\psi_0|\exp(-\beta\mathbf{H})|\psi_0\rangle \quad \det\mathbf{B}_\tau\cdots\mathbf{B}_{\Delta\tau}\mathbf{B}_\beta\cdots\mathbf{B}_{\tau+\Delta\tau} \tag{6}$$

Most importantly, both the computer time and memory requirements decrease quadratically with the density of particles for a fixed $\beta$ and $N$. In this projection framework, stabilization in the manner of (6) can be simplified in several respects and is equivalent to what Sorella, *et al.* arrive at from a different context.

Finally, while equal time measurements can always be expressed in terms of averages of the equal time Green's function $\mathbf{G}(\tau,\tau)$, calculations of susceptibilities and other time dependent quantities require matrix elements of

$$\mathbf{G}(\tau+\tau',\tau) \quad \mathbf{B}_{\tau+\tau'}\cdots\mathbf{B}_{\tau+\Delta\tau}\mathbf{G}(\tau,\tau)$$

Again writing

$$\mathbf{B}_\tau \quad \mathbf{B}_{\Delta\tau}\cdots\mathbf{U}_R\mathbf{D}_R\mathbf{V}_R$$

and

$$\mathbf{B}_\beta \quad \mathbf{B}_{\tau+\Delta\tau}\cdots\mathbf{V}_L\mathbf{D}_L\mathbf{U}_L,$$

we can calculate an unequal time Green's function stably by writing

$$\begin{aligned}\mathbf{G}(\tau,0) \quad & \mathbf{U}_R\mathbf{D}_R\mathbf{V}_R(\mathbf{I}+\mathbf{V}_L\mathbf{D}_L\mathbf{U}_L\mathbf{U}_R\mathbf{D}_R\mathbf{V}_R)^{-1}\\ & \mathbf{U}_L^{-1}(\mathbf{D}_R^{-1}\mathbf{U}_R^{-1}\mathbf{U}_L^{-1}+\mathbf{V}_R\mathbf{V}_L\mathbf{D}_L)^{-1}\mathbf{V}_R\end{aligned} \tag{7}$$

Once again we decompose the ill conditioned sum before inverting it.

## STABILITY AND ACCURACY TESTS

Various tests were performed to study the stability of these algorithms both for "toy problems" using random Hubbard Stratonovich spins and in actual simulations. Tests were performed in 32 bit, 64 bit, and 128 bit floating point precision. In testing the Green's function calculated from (3), stability requires that elements of the Green's function and measurements from simulations are invariant with machine precision. We found that matrices could be propagated stably in imaginary time up to some $\tau_0$ before decomposition is needed. $\tau_0$ depends strongly on model parameters, machine precision, and somewhat on the HS configuration, but it is insensitive to the discretization parameter $\Delta\tau$ and to the inverse temperature $\beta$. Typically, $\tau_0$ ranges from 2 to 5 and sets the limit on $\beta$ for an unstable calculation. Instability in calculations sets in suddenly as the imaginary time $\tau_0$ between decompositions is increased too much.

There is no difficulty in pushing simulations using (3) to extreme temperatures ($\beta \sim 100$) even with 32 bit floating point arithmetic. At such low temperatures we find agreement between physical quantities calculated with a finite temperature based on (3) and a ground state algorithm based on (6). We can check the results from (6) against those obtained by exact diagonalization for a small system and free fermion results for special problems. For example, for $2\times 2$ periodic Hubbard cluster with $t=1$, attractive interaction $U=-8$, and discretization parameter $\Delta\tau=0.1$, projecting the ground state from a trial wavefunction gives on site double

occupancy $< n_{p\uparrow} n_{p\downarrow} > = 0.4366$ and nearest neighbor hopping $< c_{p\sigma}^{\dagger} c_{q\sigma} > = 0.421$ at $\beta = 100$. The statistical uncertainty in the Monte Carlo results is one in the last quoted digit in each case. At $\Delta\tau = 0.05$ we obtain $< n_{p\uparrow} n_{p\downarrow} > = 0.4303$ and $< c_{p\sigma}^{\dagger} c_{q\sigma} > = 0.411$. After extrapolating these values to $\Delta\tau = 0$, we find that they agree with results from exact diagonalizations (0.4282 and 0.408) to the stated precision, which, for any finite running time, is set by the statistical fluctuations. This performance on a $2 \times 2$ cluster is typical since stability is related primarily to $\beta$ and Hamiltonian parameters that set the band edges but not to lattice size.

Preliminary tests using (5) and (7) suggest they help stabilize the simulation numerics but perhaps are not as robust as (3).

With respect to computational efficiency, we remark that in measurements[1] of the antiferromagnetic structure factor

$$\mathbf{S}(\pi,\pi) = \frac{1}{\beta N} \int_0^{\beta} \Delta\tau \quad \mathbf{M}(\tau)\mathbf{M}(\tau) \; , \qquad (8)$$

with $\mathbf{M} = \sum_p (-1)^p (n_{p\uparrow} - n_{p\downarrow})$, our finite temperature calculations for the two dimensional, half filled Hubbard model at $U/t = 4$ using (3) appear to run seven times faster at $\beta = 12$ than another calculation using an algorithm recently proposed by Hirsch.[7] Our efficiency advantage increases as the problem grows more unstable (for example, at lower temperature) since our computing time scales no worse than $(\beta/\tau_0)^2$ whereas the method of Hirsch scales as $(\beta/\tau_0)^3$.

We report elsewhere more extensive studies using these techniques.[1,10]

## ACKNOWLEDGEMENTS

We would like to thank V. Faber, T. Manteuffel, M. Parrinello, and S. Sorella for helpful discussions. This work was supported in part by the Department of Energy. One of us (E.Y.L.) would also like to thank Los Alamos National Laboratory for supporting basic research at the Laboratory.

## REFERENCES

1. S. R. White, D. J. Scalapino, R. L. Sugar, E. Y. Loh, Jr., J. E. Gubernatis, and R. T. Scalettar, "A Numerical Study of the Two Dimensional Hubbard Model with Repulsive Interaction," preprint.

2. J. E. Hirsch, *Phys. Rev. B* **31** 4403 (1985).

3. R. Blankenbecler, D. J. Scalapino, and R. L. Sugar, *Phys. Rev. D* **24** 2278 (1981).

4. R. T. Scalettar, D. J. Scalapino, R. L. Sugar and D. Toussaint, *Phys. Rev. B* **36** 8632 (1987).

5. S. Sorella, E. Tosatti, S. Baroni, R. Car, and M. Parrinello, preprint and private discussions.

6. S. R. White, R. L. Sugar, and R. T. Scalettar, "An Algorithm for the Simulation of Many Electron Systems at Low Temperatures," preprint.

7. J. E. Hirsch, "Stable Monte Carlo Algorithm for Fermion Lattice Systems at Low Temperatures," preprint.

**Recipes: The Art of Scientific Computing**. (Cambridge University Press, Cambridge, 1988); J. R. Rice, **Numerical Methods, Software, and Analysis**. (McGraw-Hill, New York, 1983).

9. The SVD decomposition proceeds by multiplying on both sides of $\mathbf{B}\cdots\mathbf{BUD}$ by orthogonal matrices; MGS multiplies only on the right side to orthogonalize the columns. Multiplication on the left side is clearly stable since it only mixes elements within the same column — elements of the same scale. Only right multiplication is tricky. For MGS, there is no difficulty since orthogonalizing a column $\mathbf{v}$ with respect to some column $\mathbf{u}$ simply entails replacing $\mathbf{v}$ with $\mathbf{v} - \mathbf{u}(\mathbf{u}^l\mathbf{v})/(\mathbf{u}^l\mathbf{u})$. Thus, independent of the relative scales of $\mathbf{u}$ and $\mathbf{v}$, one always subtracts from $\mathbf{v}$ a term which is smaller by a factor of the direction cosine between $\mathbf{u}$ and $\mathbf{v}$. For SVD, some "pivoting" is required to make the decomposition stable.

10. E. Y. Loh, Jr., J. E. Gubernatis, R. T. Scalettar, R. L. Sugar, and S. R. White, "Stable Matrix-Multiplication Methods for the Low-Temperature Simulation of Fermions," in preparation.